

# Compressed Learning for Tactile Object Classification

Brayden Hollis, Stacy Patterson, and Jeff Trinkle

**Abstract**—The potential of large tactile arrays to improve robot perception for safe operation in human-dominated environments and of high-resolution tactile arrays to enable human-level dexterous manipulation is well accepted. However, the increase in the number of tactile sensing elements introduces challenges including wiring complexity, power consumption, and data processing. To help address these challenges, we previously developed a tactile sensing technique based compressed sensing that reduces hardware complexity and data transmission, while allowing accurate reconstruction of the full-resolution signal. In this paper, we apply tactile compressed sensing to the problem of object classification. Specifically, we perform object classification on the compressed tactile data. We evaluate our method using BubbleTouch, our tactile array simulator. Our results show our approach achieves high classification accuracy, even with compression factors up to 64.

## I. INTRODUCTION

For robots to reliably and safely function in unstructured environments, they need to perceive and react to the environment. Sensors that sense a robot’s surroundings from a distance, such as vision sensors, are very useful and commonly employed, but when it comes to physical interactions, these sensors only offer a limited perspective of what is happening. Tactile sensors, sensors that sense the world through direct contact (e.g., force and temperature), greatly increase a robot’s understanding of its interactions. The concept of putting tactile sensors on a large portion of robots’ surfaces, which we refer to as *tactile skins*, has become its own research area.

There are numerous challenges with fully realizing tactile skins. To cover the various surfaces at useful resolutions requires a large number of individual sensing elements, called *taxels*, on the order of 1,000’s to 1,000,000’s [1]. Furthermore, the information needs to be gathered and processed at high rates, up to  $1kHz$  for fine force control [1]. In conflict with these desirable features, there is limited room for the wiring of these systems, especially for skins designed separately from the robot as an add-on. In addition, tactile sensors tend to be noisy, in part from contact with the environment causing misalignments.

Even with these challenges, there are a number of potential benefits that make tactile skins worth pursuing. One application of interest is tactile object recognition. Often object recognition can be performed with other sensors, such as vision, but this is not always practical or possible. For instance, a robot trying to pull items out of a crowded

cupboard may not be able to get a clear view of the item of interest due to the shelf being above or below the robot’s vision system or other objects being in the way. Using tactile sensors and tactile object recognition, the robot could find the object and retrieve it with less manipulation of the environment than grabbing items one at a time for performing visual object recognition. Another example is a rescue robot. In some disasters scenarios, visibility is limited due to smoke or dust. A rescue robot equipped with tactile sensors could still function and navigate using tactile object recognition to identify important landmarks, such as door knobs, or determine whether objects are movable or fixed, as was done by Bhattacharjee et al. in [2].

In previous work [3], we proposed using compressed sensing for data acquisition in tactile skins. Compressed sensing simultaneously samples and compresses signals. Under appropriate assumptions, the full signal can be recovered exactly or near-exactly from the compressed signal. Our approach reduces the amount of data that needs to be gathered and transferred from the tactile skin, with little to no loss of signal resolution. Fewer measurements imply increases in the signal acquisition rate, and, by performing compression in hardware, the amount of wiring can be reduced. Further, our experiments show that the reconstructed signal exhibits less noise than the raw data sampled from the noisy taxels.

In this paper, we demonstrate additional benefits of compressed sensing for tactile skins by using the compressed signals for tactile object classification. Object classification is a sub-problem of object recognition where objects are classified using a finite candidate set. We use a soft-margin support vector machine (SVM) to classify objects from their compressed tactile signals. Our tactile signals are compressed from snapshots of a tactile skin with a square array of taxels pressed onto the objects from above, similar to what might happen in the cupboard example. Direct use of the compressed signals for classification reduces processing time because the signal reconstruction phase is omitted. Further, it lowers the dimensionality of signals used for classification, which reduces processing time for both training and using the classifier. Finally, for applications where the original signal may be required after classification is performed, for example in grasping, the full tactile signals can be recovered from the compressed signals.

Tactile object classification is an active area of research. A number of works manipulate or make multiple contacts with the objects while gathering the tactile data [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. These extended interactions allow them to obtain multiple tactile perspectives of the object, which reduces uncertainty. We utilize individual tactile

B. Hollis, S. Patterson, and J. Trinkle are with the Department of Computer Science, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, USA, {hollib@rpi.edu, {sep, trink}@cs.rpi.edu}.

snapshots because for some applications, like the cupboard example, extended manipulation is not practical due to space or time constraints. Additionally, for a number of these cases, extended interaction is not necessary because most objects have a limited number of stable poses.

Jimenez et al. also use single tactile snapshots for their classification [14]. They still handle pose uncertainty, though, by pre-processing the full raw signal so all the observations are centered and oriented the same way in the tactile image. The pre-processed data is then used in a neural network for classification. We perform no pre-processing on our signals as this would be very challenging, if not impossible, in the compressed domain. Nonetheless, we achieve high classification accuracy.

Previous works also differ in the types of data used for classification. Some approaches use the full raw tactile data [14], [11], [13]. A number of the methods reduce the dimensionality of the data before, or as part of, the machine learning process. Some manually reduce the dimension by selecting features such as average force and contact area [7], [2], [15]. Others use automated methods to reduce the dimension, for instance, Self-Organizing Maps [8], [10] and Principle Component Analysis [5], [6], [12]. In our approach, the dimension of the data is reduced as part of the acquisition process, which may be done in hardware [16]. This dimension reduction is not a pre-processing step.

The rest of the paper is organized as follows. We give a general overview of SVMs in Section II. We review compressed sensing theory and our work in applying it to tactile skins in Section III. Section IV describes compressed learning theory and our application of it to tactile skins. Our experiments and data are explained in Section V and VI, respectively. We present our results in Section results, and in Section VIII, we conclude with some final thoughts and ideas for future work.

## II. BACKGROUND

Support Vector Machines are a popular classification tool in machine learning. They classify the observations by finding a hyperplane that separates the training data into its two classes such that the distance between the hyperplane and the closest observations is maximized [17]. This distance is called the *margin* and the closest observations are called *support vectors*, as they are the observations that determine the classifier. To find the hyperplane, one solves the following minimization problem,

$$\begin{aligned} & \underset{b, \mathbf{w}}{\text{minimize}} && \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ & \text{subject to} && \ell_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \\ & && \text{for } i = 1, \dots, N, \end{aligned} \quad (1)$$

where  $N$  is the number of observations;  $\mathbf{x}_i \in \mathbb{R}^n$ , for  $i = 1$  to  $N$ , are the observations;  $\ell_i \in \{-1, 1\}$ , for  $i = 1$  to  $N$ , are the class labels;  $\mathbf{w}$  is a vector orthogonal to the separating hyperplane; and  $b = -\mathbf{w}^T \mathbf{x}_0$  for any point  $\mathbf{x}_0$  on the hyperplane. This optimization problem is a convex quadratic program, a well studied class of problems with

many implemented solvers. Once (1) is solved for  $\mathbf{w}^*$  and  $b^*$ , to classify a point  $\hat{\mathbf{x}}$ , one simply solves  $\text{sign}(\mathbf{w}^{*T} \hat{\mathbf{x}} + b)$ .

Often it is not possible to completely separate the data with a hyperplane, so SVMs can be modified to allow some observations to be misclassified as follows:

$$\begin{aligned} & \underset{b, \mathbf{w}}{\text{minimize}} && \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ & \text{subject to} && \ell_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ & && \xi_i \geq 0 \\ & && \text{for } i = 1, \dots, N, \end{aligned} \quad (2)$$

where  $\xi_i$  is the amount the  $i$ th observation violates the margin, known as the *hinge loss*, and  $C$  is a parameter to balance between maximizing the margin and reducing the margin violations. This modification is known as soft-margin SVMs.

When training soft-margin SVMs, cross-validation is used to tune the parameter  $C$ . Cross-validation separates the *training* set into a *development* set and a *validation* set. The development set is used to train multiple SVMs. All the trained models are then evaluated with the validation set. The model that performs the best is then retrained using the full training set and is the final learned classifier.

The SVMs discussed so far are binary classifiers. Some modification is necessary for multi-class classification. There are various extensions (for an overview see [18]), of which we use the Direct Acyclic Graph SVM (DAGSVM). DAGSVM trains a binary SVM for each pair of classes [19]. During classification, an observation is classified by a binary SVM. That observation is then classified in another binary SVM with the previously assigned class as one of the two potential classes and the other class eliminated from future consideration. This is continued, sequentially eliminating classes from consideration, until a single class remains. The observation is classified as an element of the remaining class. Thus for a  $M$ -class problem, DAGSVM trains  $\frac{M(M-1)}{2}$  binary SVMs, but classifies an observation only using  $M-1$  SVMs. It was found that the order in which the binary SVMs are used for classifying the observations does not matter [19].

While maximizing classification accuracy (the percent of classes correctly labeled) is the true objective of SVMs, SVMs can also be evaluated by the expected hinge loss  $H_D(\mathbf{w}^+)$  over the problem distribution  $D$ , where  $\mathbf{w}^+ \in \mathbb{R}^{n+1}$  is the vector generated by concatenating  $\mathbf{w}^*$  and  $b^*$ . More formally,

$$H_D(\mathbf{w}^+) = \mathbb{E}_D \left[ \max\{0, 1 - y(\mathbf{w}^{*T} \mathbf{x} + b^*)\} \right]. \quad (3)$$

$\mathbb{E}_D[\cdot]$  is the expectation over  $D$ .

## III. COMPRESSED SENSING FOR TACTILE SKINS

### A. Compressed Sensing Theory

In compressed sensing, a signal, for example, force readings from a tactile array at a given time, is simultaneously measured and compressed by taking linear combinations of the signal components. Specifically, the *compressed signal*

$\mathbf{y} \in \mathbb{R}^m$  with elements  $y_i$  is obtained from the *full signal*  $\mathbf{x} \in \mathbb{R}^n$  as follows:

$$\mathbf{y} = \Phi \mathbf{x}, \quad (4)$$

where  $\Phi = [\Phi_{ij}]$  is the  $m \times n$  *measurement matrix*. If  $m < n$ , (4) is under-determined and, in general,  $\mathbf{x}$  cannot be recovered.

In compressed sensing, one considers a restricted set of signals that are sparse in some representation basis. Formally, a signal  $\mathbf{x}$  is  $k$ -sparse in a representation basis  $\Psi \in \mathbb{R}^{n \times n}$  if there is a  $k$ -sparse vector  $\mathbf{s}$ , meaning  $\mathbf{s}$  has at most  $k$  non-zero entries, such that  $\mathbf{x} = \Psi \mathbf{s}$ . Compressed sensing theory provides conditions under which such sparse vectors can be recovered from fewer than  $n$  measurements. One such condition relates to the *restricted isometry property*, which is defined as follows.

*Definition 1:* A matrix  $A$  satisfies the  $k$ -restricted isometry property ( $k$ -RIP) if there exists a  $\delta \in (0, 1)$  such that

$$(1 - \delta) \|\mathbf{s}\|_2^2 \leq \|A\mathbf{s}\|_2^2 \leq (1 + \delta) \|\mathbf{s}\|_2^2, \quad (5)$$

holds for all  $k$ -sparse  $\mathbf{s}$ .

If  $\mathbf{x}$  is  $k$ -sparse in a representation basis  $\Psi$ , and the matrix  $\Phi\Psi$  satisfies the  $2k$ -RIP, then  $\mathbf{x}$  can be recovered exactly from  $m \in O(k \log n)$  measurements [20]. Many efficient recovery algorithms have been proposed. Further, it has been shown robust recovery in the presence of noise in the signal and/or measurements is feasible [20].

### B. Application to Tactile Skins

In previous work [3], we investigated the application of compressed sensing in planar tactile arrays. For the measurement matrix, we used the Scrambled Block Hadamard Ensemble (SBHE), which was developed by Gan et al. [21] for compressed sensing in the image domain. SBHE is a partial block Hadamard transform with randomly permuted columns and can be represented as

$$\Phi_H = Q_m W P_n, \quad (6)$$

where  $W$  is a  $n \times n$  block diagonal matrix with each block a  $B \times B$  Hadamard matrix.  $P_n$  is the permutation matrix, which randomly reorders the  $n$  columns of  $W$ , and  $Q_m$  selects  $m$  rows of  $W P_n$  uniformly at random. The SBHE was selected for its potential hardware implementations since it separates the sensors into disjoint measurement groups. This allows simpler wiring, as each measurement samples only a limited number of taxels, and measurements within a group could share wiring. Additionally, different groups could be measured in parallel. The number of elements in a group is equal to  $B$ , which we set to 32 in this work.

Figure 1(a) demonstrates the wiring on a small  $3 \times 3$  array. There are three measurement groups of three sensors each. The sensors in a group are daisy-chained together, with any measurement of that group able to use the same wire path. For comparison, Figure 1(b) shows the same array with each element wired separately for individual taxel readings.

We used the Daubechies-2 wavelet transform [22] for our basis  $\Psi$ . This transform is similar to the wavelet transforms

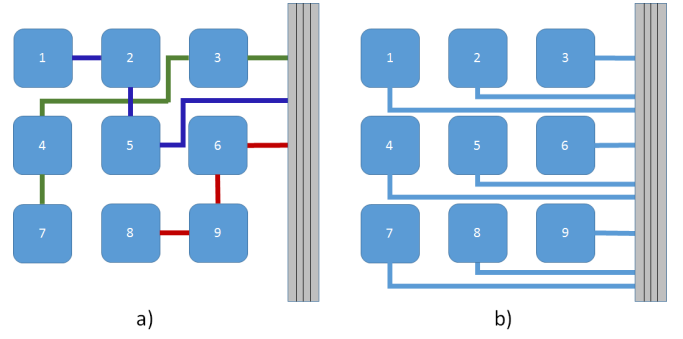


Fig. 1: Example wiring schematics for a) compressed sensing measurements on a  $3 \times 3$  tactile grid and b) individual sensor measurements.

used in image compression such as JPEG2000. A similar Daubechies wavelet transform was also used in Gan et al.'s experiments [21].

Using these standard (i.e. pre-existing and non-optimized) compressed sensing tools, we achieved 50Hz reconstruction rates for a tactile array of 4096 taxels from 1365 measurements (a compression factor of 3) [3]. This array contains approximately the same number of sensors as the the largest existing tactile system, and the reconstruction rate is on the same order of magnitude as that system's measurement rate [23]. In addition, the reconstructed signal had less noise than the raw signal, and the system has potential for wire reduction.

## IV. COMPRESSED LEARNING FOR TACTILE SKINS

We propose a technique for applying compressed sensing to tactile object recognition. Specifically, we perform object classification on the compressed signals. This approach maintains the benefits of compressed sensing described above. In addition, it reduces processing time by avoiding signal reconstruction, and it also reduces the dimension of the signals used for classification.

We first briefly review the theory of classification using compressed signals and then give details of our method.

### A. Compressed Learning

Calderbank et al. first proposed classification using compressed signals, a technique that they have called *compressed learning* [24]. Specifically, they show that with high probability, a soft-margin SVM trained on compressed signals has expected accuracy similar to the best linear classifier in the uncompressed data domain. This is formalized in the following theorem.

*Theorem 1 (Thm. 3.1 [24]):* Let  $D$  be a distribution of  $k$ -sparse vectors  $\mathbf{x}_i \in \mathbb{R}^n$  such that for all  $i$ ,  $\|\mathbf{x}_i\|_2 \leq R$ , where  $R$  is a known upper bound. Further, assume that for each  $\mathbf{x}_i$  there is a label  $\ell_i \in \{-1, 1\}$ . Let  $\Phi \in \mathbb{R}^{m \times n}$  be a measurement matrix that satisfies  $2k$ -RIP with constant  $\delta$ . Additionally, let

$$S_\Phi = \{(\Phi \mathbf{x}_1, \ell_1), \dots, (\Phi \mathbf{x}_N, \ell_N)\}$$

be i.i.d. labeled instances compressively sampled from  $D$ , and let  $\mathbf{z}_{S_\Phi} \in \mathbb{R}^m$  be the linear classifier from the soft-margin SVM trained on  $S_\Phi$ . Finally, let  $\mathbf{w}_0 \in \mathbb{R}^n$  be the best linear classifier in the uncompressed data domain with low expected hinge loss over  $D$ ,  $H_D(\mathbf{w}_0)$ , and large margin (hence small  $\|\mathbf{w}_0\|_2$ ). Then with probability  $1 - 2\rho$  over  $S_\Phi$ :

$$H_D(\mathbf{z}_{S_\Phi}) \leq H_D(\mathbf{w}_0) + O\left(\|\mathbf{w}_0\|_2 \left(R^2\delta + \frac{\log(\frac{1}{\rho})}{N}\right)^{\frac{1}{2}}\right). \quad (7)$$

In (7),  $R$ ,  $\mathbf{w}_0$ , and  $\rho$  are fixed by the problem. The variable  $\delta$  is as defined in Definition 1 and typically increases as the amount of compression increases. Thus, (7) implies greater compression leads to less confidence in the classifier’s accuracy, which is to be expected. Also, as expected, the accuracy depends on training set size  $N$ . As  $N$  increases, the accuracy of the compressed classifier approaches that of the optimal linear classifier.

Theorem 1 addresses classification with compressed signals obtained by measuring sparse signals. If the signals are themselves not sparse, but are sparse in some orthonormal basis  $\Psi$ , then Theorem 1 still applies, provided  $\Phi\Psi$  satisfies  $2k$ -RIP [24]. An important point to note is that, unlike in compressed sensing, for compressed learning it is not necessary for this basis to be known.

### B. Application to Tactile Skins

We propose to use compressed learning for object classification in tactile skins. We generate each observation as follows. The full signal is compressed using the SBHE matrix, described in Section III-B, to generate the compressed signal of a single time instance of contact with an object. More formally, for each observation, let  $\mathbf{x}_i$  represent an  $n$ -vector containing the sensor readings of all taxels (which may be noisy) at the measurement time for contact with a single object. We generate a single compressed signal, an  $m$ -vector, from  $\mathbf{x}_i$  using the SBHE, i.e.,  $\mathbf{y}_i = \Phi\mathbf{x}_i$ .

For classification, we use the soft-margin DAGSVM described in Section II. We use the DAGSVM implemented in the MATLAB SVM Toolbox from University of East Anglia [25] with a validation set to perform a grid search for the parameter  $C$  in (2).

Gan et al. [21] prove that for many basis matrices  $\Psi$  that have applications in image compression, the product  $\Phi\Psi$  behaves like a Gaussian i.i.d. matrix. Further, it has been shown a Gaussian i.i.d. matrix with  $m \in O(k \log(n/k)/\delta^2)$  rows satisfies the  $k$ -RIP with high probability [26]. Therefore, by Theorem 1, we should obtain similar classification performance on compressed signals as we would on the full signals.

We validate this approach through simulations, which we detail in the following sections.

## V. EXPERIMENTAL SETUP

We generated tactile array data using our BubbleTouch simulator (<https://github.com/bdhollis/BubbleTouch>). BubbleTouch represents taxels as rigid spheres suspended in



Fig. 2: A union-of-spheres model generated from the Yale-CMU-Berkeley (YCB) drill.

space by spring and damper pairs; one pair per sphere. Contact interactions between objects and taxels are assumed to be quasistatic to avoid simulation instabilities that commonly arise in the simulation of dynamic contact models. Tactile arrays of any shape and distribution can be created simply by creating a rigid substrate body with that shape and attaching the bases of the springs to it in the desired pattern. To explore the effect of array resolution, we created eight planar square grid arrays with the same overall dimensions ( $256\text{mm}$  by  $256\text{mm}$ ), but different resolutions ranging from  $4\text{mm}$  taxels in a  $64 \times 64$  array, to  $256\text{mm}$  taxels in a  $1 \times 1$  array.

Our objects were obtained from two sources: the YCB (Yale-CMU-Berkeley) Object Set [27] and a few simple geometric shapes. To simplify collision detection, the objects were approximated as union of spheres. To convert a YCB object to a union-of-spheres model, we used the UC Berkley Poisson reconstructed mesh vertices as the sphere centers. For each object, all the spheres were assigned a radius equal to twice the mean distance between all vertices and their nearest neighboring vertex. An example of a union-of-spheres model for the YCB drill is shown in Figure 2. For primitives, for example, ellipsoids, boxes, and cylinders, we manually designed the union-of-spheres models.

To generate tactile data, each object was placed in a stable configuration on a rigid horizontal support plane and touched from above by the tactile array. Each touch was performed by initially positioning the substrate of the array far enough above the object to avoid contact. From there, the substrate was translated downward, causing the taxels to contact the object and move relative to the substrate (deforming the springs and dampers). After a pre-specified downward motion of the substrate, the simulation was stopped and the spring and

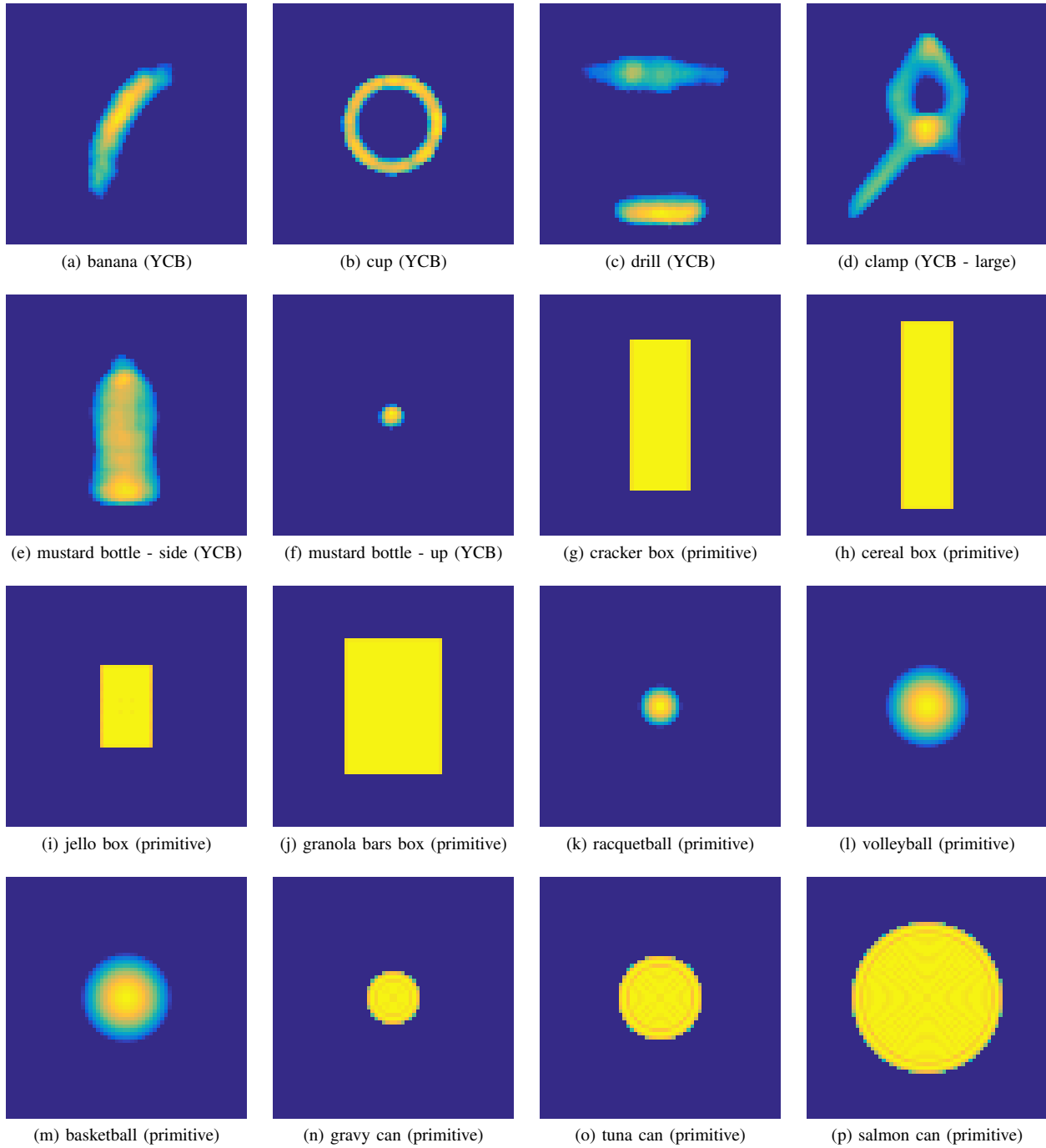


Fig. 3: Example noiseless tactile image for each object. In parentheses next to the object name is the object model source. The color scale goes from dark blue to yellow, where dark blue is no contact and yellow is contact of  $0.01N$



damper forces at all the taxels were taken as the noiseless tactile signal for this observation. To approximate the noise in real tactile sensors, random zero-mean Gaussian noise with standard deviation of  $0.001N$  (equal to 5% of the signal range) was added to each taxel reading. Values outside a taxel's range ( $[0, 0.02]N$ ) were clipped to the boundary. The array signal with added noise was taken as the *raw signal*. Figure 3 shows 16 images gathered from 15 objects (the mustard bottle was touched in two different orientations). The sources of the object models are in parentheses next to their names.

When testing our compressed learning algorithm, we started with the raw signal from the array of the finest resolution, i.e.,  $64 \times 64$ . The uncompressed signal was thus of length 4,096. For each raw signal, we generated a *compressed signal* of length  $m$  by left multiplying the raw signal by the appropriately-sized SBHE matrix. We use the following values for  $m$ : 1,024, 256, 64, 16, 4, and 1.

For comparison, we also performed classification using raw signals of the same dimensions as the compressed signals. We generated raw signals using coarser tactile arrays, consisting of 1,024, 256, 64, 16, 4, and 1 taxels. For convention, we use *signal size* to refer to the number of elements in a signal's vector. This means for compressed signals, the signal size is  $m$ , the number of measurements, and for the raw signals, the signal size is the number of taxels in the array.

## VI. DATA SETS FOR CLASSIFICATION

For each of the 16 objects, 360 touching observations were done as described above, but with systematic off-sets from the nominal starting configuration of the array. The off-sets were  $(0, 2, 4, 6, 8, 10)mm$  along the rows of the array,  $(0, 2, 4, 6, 8, 10)mm$  along the columns of the array, and  $(0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ, 40^\circ, 45^\circ)$  rotations about an axis normal to the array. This yielded 5,760 observations.

The development set was formed by choosing 40% of the 360 perturbations uniformly at random. All observations with those 144 perturbations for all 16 objects defined the development set, which contained 2,304 observations. The validation set was formed similarly, using 20% of the other observations. The test set was defined as the remaining 2,304 observations. We also tested development sets using 20%, 10%, 6%, 2%, and 0.67% of the 360 perturbations with the corresponding validation sets of 10%, 5%, 3%, 1%, and 0.33%.

## VII. RESULTS

Figure 4 shows the overall classification accuracies for seven different signal sizes. In addition to the accuracy rates for the compressed signals, we also look at the accuracy rates for the raw signals of corresponding dimensions. Figure 4 also shows the results for two different training set sizes of 60% and 3%. All the results are averages over 10 different splits of the data set into test and training tests.

Smaller signal sizes yield less accurate classification, but even with fairly small signal sizes, the classification has a

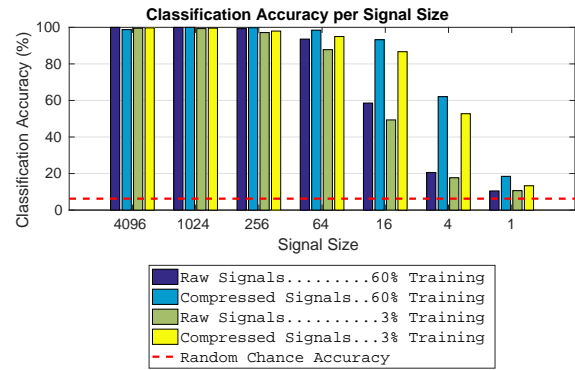


Fig. 4: The classification accuracy for various signal sizes using two training set sizes. For the raw signals, signal size refers to the number of taxels in the array, and for the compressed signals, signal size refers to the number of measurements. The dotted line is the accuracy of randomly assigning a label to each example.

high success rate. The raw signals achieve over 85% accuracy even at a signal size of 64, with 93.3% and 87.8% for training sets of 60% and 3% respectively. The compressed signals achieve that level of accuracy for the signal size of 16, a compression factor of 256, with 93.2% and 86.7% respective accuracies. Overall, the compressed signals outperformed the corresponding raw signals of the same size. The exception is for the signal size of 4,096 for which there is no compression. These results agree with Theorem 1. The compressed signals have similar accuracies to the classifier on the original signal (the raw signal of size 4096). Further, the accuracy deviates more with increased compression.

An interesting item to note is compressed signals of size less than 128 do not use every taxel value, and for the signals of size 64, at most half of the taxels are actually used. Compressed signals of larger size also may not involve every taxel.

Collecting large sets of training data can be inconvenient, challenging, or impractical in deployed hardware. So, we explored performance with respect to various amounts of training data to determine how much is needed for accurate classification. Our results are shown in Figure 5. It shows the classification rates for the raw signals of size 4,096, compressed signals obtained from this raw signal, of sizes 1,024 and 16, and raw signals obtained from coarser tactile arrays of sizes 1,024 and 16. Again, the results are averaged over 10 splits of the data set.

The three larger signals all perform extremely well, with near 100% accuracy for all but the smallest training set size of 1%. Even at the 1% training set size, the three signals achieve over 90% success. For the smaller signals, the accuracy decreases as the amount of training data decreases. The raw signals of size 16 have under 60% accuracy for all training sizes. The compressed signals of size 16 perform much better, maintaining over 80% accuracy even with 3% training data. At 1% training data, the smaller-sized com-

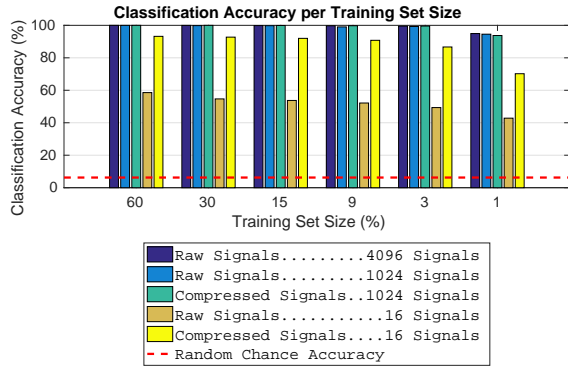


Fig. 5: The classification accuracy for training sets of various sizes. For the raw signals, the number of signals refers to the number of taxels in the array, and for the compressed signals, the number of signals refers to the number of measurements. The dotted line is the accuracy of randomly assigning a label to each example.

pressed signal classification accuracy drops to approximately 70%.

As was the case for the results shown in Figure 4, these results also agree with compressed learning theory. The classification accuracies for the compressed signals continue to be similar to the results from the original signal, but the deviation increases as the training set size decreases in accordance with (7). This is most clearly seen in Figure 5 by comparing the deviations between original signal (raw signal of size 4,096) and the compressed signal of size 16. The deviation goes from approximately 7% to 15% between 60% and 1% training set sizes.

To get a better understanding of how the classifier is performing between individual classes, we computed the confusion matrix, which shows the percentage of observations of each class that are labeled as a particular class. This helps to identify which pairs of classes are hard to discriminate. Figure 6 shows the confusion matrix for the compressed signals of size 64, trained on 3% of the observations per object, averaged over the ten splits of the data-set. From the strong diagonal it is clear the classification performs well overall. The greatest confusion occurs between the volleyball and the basketball; approximately 25% of the time one is mistaken for the other. This is understandable because both are spheres with similar radii and similar tactile signals, as seen in Figures 3(l) and 3(m). The gravy can and the volleyball also generate a bit of confusion. While this is less intuitive, it is not surprising. Both objects have round shapes, and while the volleyball has a much larger radius overall, Figures 3(l) and 3(n) show the contact radii are similar. There is also a little confusion between the upright mustard bottle and the racquetball since they also have circular contacts of similar radii. The other confusions of note is classifying the cereal box as either the cracker box or the mustard bottle on its side and the jello box as the basketball. This is a little less apparent, but the shape and dimensions are similar between

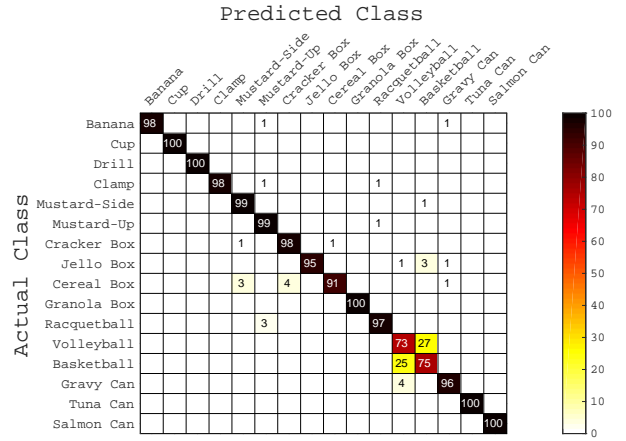


Fig. 6: The average confusion matrix over the 10 data-set splits for compressed signals of 64 elements trained on three percent of the examples of each object. The values are the percentage of the actual class examples that were label as the predicted class. Locations with no stated values have approximately zero percent of the actual class labeled as the predicted class.

the cereal box and the other two items, and basketball covers similar area as the jello box.

## VIII. CONCLUSION

We have developed and demonstrated an approach for tactile object classification using compressed learning. Our approach classified various objects with high accuracy, even with high levels of compression and small amounts of training data. The compressed signals generally resulted in performance similar to the full raw signal of individual taxel readings and outperformed raw signals of corresponding signal size.

Our approach offers benefits of reduced data acquisition and processing time, as well as the potential to reduce wiring complexity in hardware implementations. In addition, for tasks where the full raw signal is required after classification, this signal can be recovered from the compressed signal.

In future work, we will explore the application of compressed sensing to other tactile tasks, such as object manipulation, safe interaction with humans, and robot locomotion through rough terrain. In addition, we will investigate other compression techniques and wiring configurations. Finally, we plan to implement our approach in hardware to fully evaluate its benefits.

## REFERENCES

- [1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile Sensing: From Humans to Humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, Feb 2010.
- [2] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp, "Haptic Classification and Recognition of Objects Using a Tactile Sensing Forearm," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2012, pp. 4090–4097.
- [3] B. Hollis, S. Patterson, and J. Trinkle, "Compressed Sensing for Tactile Skins," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 150–157.

- [4] R. Russell, "Object recognition by a 'smart' tactile sensor," in *Proc. Aust. Conf. Robot. Autom.*, 2000, pp. 93–98.
- [5] G. Heidemann and M. Schöpfer, "Dynamic Tactile Sensing for Object Identification," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2004, pp. 813–818 Vol.1.
- [6] M. Schöpfer, H. Ritter, and G. Heidemann, "Acquisition and Application of a Tactile Database," *Proc. IEEE Int. Conf. Robot. Autom.*, no. April, pp. 1517–1522, 2007.
- [7] M. Schöpfer, M. Pardowitz, and H. Ritter, "Using Entropy for Dimension Reduction of Tactile Data," *Proc. Int. Conf. Adv. Robot.*, pp. 1–6, 2009.
- [8] S. Takamuku, A. Fukuda, and K. Hosoda, "Repetitive Grasping with Anthropomorphic Skin-Covered Hand Enables Robust Haptic Recognition," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3212–3217, 2008.
- [9] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object Identification with Tactile Sensors Using Bag-of-Features," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 243–248, 2009.
- [10] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, "Haptic Object Recognition Using Passive Joints and Haptic Key Features," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2349–2355.
- [11] K. Hosoda and T. Iwase, "Robust Haptic Recognition by Anthropomorphic Bionic Hand through Dynamic Interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2010, pp. 1236–1241.
- [12] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager, "Tactile-Object Recognition from Appearance Information," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 473–487, 2011.
- [13] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano, "Tactile Object Recognition Using Deep Learning and Dropout," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2014, pp. 1044–1050.
- [14] A. Jiménez, a.S. Soembagijo, D. Reynaerts, H. Van Brussel, R. Ceres, and J. Pons, "Featureless Classification of Tactile Contacts in a Gripper using Neural Networks," *Sensors Actuators A Phys.*, vol. 62, no. 1-3, pp. 488–491, 1997.
- [15] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic, "Design of a Flexible Tactile Sensor for Classification of Rigid and Deformable Objects," *Robot. Autom. Syst.*, vol. 62, no. 1, pp. 3–15, 2014.
- [16] M. F. Duarte, Y. C. Eldar, and S. Member, "Structured Compressed Sensing : From Theory to Applications," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, 2011.
- [17] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining to Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [18] C. W. Hsu and C. J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [19] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification," *Adv. Neural Inf. Process. Syst.*, pp. 547–553, 2000.
- [20] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.
- [21] L. Gan, T. Do, and T. Tran, "Fast compressive imaging using scrambled block hadamard ensemble," in *Proc. 16th European Signal Processing Conference*, Aug 2008, pp. 1–5.
- [22] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.
- [23] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta, "The Design of the iCub Humanoid Robot," *Int. Jour. Humanoid Robot.*, vol. 9, no. 04, 2011.
- [24] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed Learning: Universal Sparse Dimensionality Reduction and Learning in the Measurement Domain," 2009. [Online]. Available: <http://dsp.rice.edu/files/cs/cl.pdf>
- [25] G. C. Cawley, "MATLAB support vector machine toolbox (v0.55 $\beta$ )," University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000. [Online]. Available: <https://www.uea.ac.uk/computing/matlab-svm-toolbox>
- [26] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.
- [27] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015. [Online]. Available: <http://www.ycbbenchmarks.org/>